

[illegible]

3

Sy

MT

MT
MT

MT

MT

MT
MTMT
MTMT
MTMT
MT

MT
MT

MT
MT

MT

MT

MT
MT

MI
MT

MT
MT

MT
MT

MT
MT

MT
MT

MT

111

227

MT

MT
MT

MT

MT

M1
M1MT
MT

M1

M1

10

1

10

100

10

1

```

LL               IIIIII               SSSSSSSS
LL               IIIIII               SSSSSSSS
LL               II                    SS
LL               II                    SS
LL               II                    SS
LL               II                    SS
LL               II                    SSSSSS
LL               II                    SSSSSS
LL               II                    SS
LL               II                    SS
LL               II                    SS
LL               II                    SS
LLLLLLLLLLLLLL  IIIIII               SSSSSSSS
LLLLLLLLLLLLLL  IIIIII               SSSSSSSS

```

(2)	54	HISTORY ; Detailed Current Edit History
(3)	77	DECLARATIONS ; Declarative Part of Module
(4)	113	MTH\$GTAN - Standard G Floating GTAN
(5)	156	MTH\$GTAN_R7 - JSB entry point
(6)	201	MTH\$GTAND - Standard G Floating GTAND
(7)	244	MTH\$GTAND_R7 - JSB entry point


```
0000 1      .TITLE  MTH$GTAN      ; G Floating Point Tangent routine
0000 2      ; (GTAN,GTAND)
0000 3      .IDENT /1-011/      ; File: MTHGTAN.MAR  EDIT: SBL1011
0000 4      :
0000 5      :*****
0000 6      :
0000 7      :*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8      :*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9      :*  ALL RIGHTS RESERVED.
0000 10     :
0000 11     :*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12     :*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13     :*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14     :*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15     :*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16     :*  TRANSFERRED.
0000 17     :
0000 18     :*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19     :*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20     :*  CORPORATION.
0000 21     :
0000 22     :*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23     :*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24     :
0000 25     :
0000 26     :*****
0000 27     :
0000 28     :
0000 29     : FACILITY: MATH LIBRARY
0000 30     :++
0000 31     : ABSTRACT:
0000 32     :
0000 33     : MTH$GTAN is a function which returns the G floating point tangent
0000 34     : of its G floating point radian argument. The call is standard
0000 35     : call-by-reference. It JSB to MTH$GTAN_R7.
0000 36     :
0000 37     : MTH$GTAND is a function which returns the G floating point tangent
0000 38     : of its G floating point degree argument. The call is standard
0000 39     : call-by-reference. It JSB to MTH$GTAND_R7.
0000 40     :
0000 41     :--
0000 42     :
0000 43     : VERSION: 1
0000 44     :
0000 45     : HISTORY:
0000 46     : AUTHOR:
0000 47     :      Steven B. Lionel, 12-Jan-79: Version 1
0000 48     :
0000 49     : MODIFIED BY:
0000 50     :
0000 51     :
0000 52     :
```

```
0000 54      .SBTTL HISTORY ; Detailed Current Edit History
0000 55
0000 56
0000 57 : Edit History for Version 1 of MTH$GTAN
0000 58 :
0000 59 : 1-001 - Adapted from MTH$DTAN version 1-002. SBL 15-Jan-79
0000 60 : 1-002 - Handle exceptions correctly and use general addressing for
0000 61 : externals. JBS 16-AUG-1979
0000 62 : 1-003 - Add JSB entry point. JBS 16-AUG-1979
0000 63 : 1-004 - Correct a typo in edit 003. JBS 17-AUG-1979
0000 64 : 1-005 - Do range check in JSB routine with CMPZV rather than with a pair of
0000 65 : CMPG's. JAW 16-Oct-1979
0000 66 : 1-006 - Change MOVD to MOVG in JSB routine. JAW 17-Oct-1979
0000 67 : 1-007 - Have CALL use JSB routine. Fix error handling.
0000 68 : SBL 31-Oct-1979
0000 69 : 1-008 - Added degree entry points. RNH 8-MAR-1981
0000 70 : 1-009 - Modify MTH$GTAN_R7 and MTH$GTAND_R7 to JSB to MTH$GSINCOS_R7 and
0000 71 : MTH$GSINCOSD_R7 instead of MTH$GSIN_R7, MTH$GCOS_R7, and
0000 72 : MTH$GSIND_R7, MTH$GCOSD_R7. RNH 27-AUG-81.
0000 73 : 1-010 - Change shared external references to G^ RNH 25-Sep-81
0000 74 : 1-012 - Use general mode addressing. SBL 30-Nov-1981
0000 75 :
```



```
0000 77      .SBTTL  DECLARATIONS      ; Declarative Part of Module
0000 78
0000 79 :
0000 80 : INCLUDE FILES:
0000 81 :
0000 82 : EXTERNAL SYMBOLS:
0000 83      .DSABL  GBL                ; prevent undefineds from becoming
0000 84                                     ; global
0000 85      .EXTRN  MTH$GSINCOS_R7      ; G Floating sine cosine routine (radian)
0000 86      .EXTRN  MTH$GSINCOSD_R7     ; G Floating sine cosine routine (degree)
0000 87      .EXTRN  MTH$$SIGNAL         ; Math error signal routine
0000 88      .EXTRN  MTH$K_FLOOVEMAT     ; Overflow error code
0000 89      .EXTRN  MTH$K_FLOUNDMAT    ; Underflow error code
0000 90      .EXTRN  MTH$$JACKET_TST
0000 91 :
0000 92 : EQUATED SYMBOLS:      none
0000 93 :
0000 94 : MACROS:
0000 95      $$FDEF                    ; Define SF (Stack Frame) symbols
0000 96 :
0000 97 :
0000 98 : PSECT DECLARATIONS:
0000 99
00000000 100      .PSECT  _MTH$CODE      PIC,SHR, LONG, EXE, NOWRT
0000 101                                     ; program section for math routines
0000 102 :
0000 103 : OWN STORAGE:      none
0000 104 :
0000 105 : CONSTANTS:
0000 106
00000004 107      X = 4                                ;Position of argument from AP.
0000 108
0000 109 G_SMALLEST DEG:
C1F81A63 A5DC006C 0000 110      .LONG  ^XA5DC006C, ^XC1F81A63 ; 180/pi*2**1024
0008 111
```

```
0008 113 .SBTTL MTH$GTAN - Standard G Floating GTAN
0008 114
0008 115
0008 116 :++
0008 117 : FUNCTIONAL DESCRIPTION:
0008 118 :
0008 119 : GTAN - G floating point function
0008 120 :
0008 121 : For algorithm, see MTH$GTAN_R7.
0008 122 :
0008 123 : CALLING SEQUENCE:
0008 124 :
0008 125 : gtan.wg.v = MTH$GTAN(X.rg.r)
0008 126 :
0008 127 : INPUT PARAMETERS:
0008 128 :
0008 129 : X.rg.r ;Address of value of angle in radians.
0008 130 :
0008 131 : IMPLICIT INPUTS: none
0008 132 :
0008 133 : OUTPUT PARAMETERS:
0008 134 :
0008 135 : VALUE: G floating tangent of the argument
0008 136 :
0008 137 : IMPLICIT OUTPUTS: none
0008 138 :
0008 139 : COMPLETION CODES: none
0008 140 :
0008 141 : SIDE EFFECTS:
0008 142 :
0008 143 : See MTH$GTAN_R7
0008 144 :
0008 145 : ---
0008 146
0008 147
40FC 0008 148 .ENTRY MTH$GTAN, ^M<IV, R2, R3, R4, R5, R6, R7>
000A 149 ; standard call-by-reference entry
000A 150 ; disable DV (and FU), enable IV
000A 151
MTH$FLAG_JACKET
6D 00000000'GF 9E 000A
0011
0011
0011
50 04 BC 50FD 0011 152 MOVG @X(AP), R0 ; R0/R1 = argument
01 10 0016 153 BSBB MTH$GTAN_R7 ; Get the tangent
04 0018 154 RET ; With result in R0-R1
```



```
0019 156 .SBTTL MTH$GTAN_R7 - JSB entry point
0019 157 :
0019 158 :++
0019 159 : FUNCTIONAL DESCRIPTION:
0019 160 :
0019 161 : GTAN - JSB entry point
0019 162 :
0019 163 : Algorithmic steps:
0019 164 : 1. Compute GSIN and GCOS.
0019 165 : 2. If GCOS is zero, we have an error.
0019 166 : 3. Return GSIN / GCOS.
0019 167 :
0019 168 : CALLING SEQUENCE:
0019 169 :
0019 170 :     MOVG    argument, R0
0019 171 :     JSB     MTH$GTAN_R7
0019 172 :
0019 173 : INPUT PARAMETERS:
0019 174 :
0019 175 :     R0 / R1 contains x
0019 176 :
0019 177 : IMPLICIT INPUTS:
0019 178 :
0019 179 :     NONE
0019 180 :
0019 181 : OUTPUT PARAMETERS:
0019 182 :
0019 183 :     The result is the tangent of x, in G_floating.
0019 184 :
0019 185 : IMPLICIT OUTPUTS:
0019 186 :
0019 187 :     NONE
0019 188 :
0019 189 : SIDE EFFECTS:
0019 190 :
0019 191 :     NONE
0019 192 : --
0019 193 MTH$GTAN_R7::
0019 194 JSB     G^MTH$GSINCOS_R7
0019 195 TSTG    R2
0022 196 BEQL    COSZER
0024 197 DIVG2   R2, R0
0028 198 RSB
0029 199

00000000'GF 16
          52 53FD
          69 13
50        52 46FD
          05
```

```
: entry point
: Compute GSIN(X), and GCOS(X).
: Is GCOS(X) zero ?
: If zero, error
: Compute GSIN / GCOS
: Return.
```



```
0029 201      .SBTTL MTH$GTAND - Standard G Floating GTAND
0029 202
0029 203
0029 204      :++
0029 205      : FUNCTIONAL DESCRIPTION:
0029 206      :
0029 207      : GTAND - G floating point function
0029 208      :
0029 209      :     For algorithm, see MTH$GTAND_R7.
0029 210      :
0029 211      : CALLING SEQUENCE:
0029 212      :
0029 213      :     GTAND.wg.v = MTH$GTAND(x.rg.r)
0029 214      :
0029 215      : INPUT PARAMETERS:
0029 216      :
0029 217      :     X.rg.r                                ;Address of value of angle in degrees.
0029 218      :
0029 219      : IMPLICIT INPUTS:      none
0029 220      :
0029 221      : OUTPUT PARAMETERS:
0029 222      :
0029 223      :     VALUE: G floating tangent of the argument
0029 224      :
0029 225      : IMPLICIT OUTPUTS:     none
0029 226      :
0029 227      : COMPLETION CODES:    none
0029 228      :
0029 229      : SIDE EFFECTS:
0029 230      :
0029 231      :     See MTH$GTAND_R7
0029 232      :
0029 233      : ---
0029 234
0029 235
40FC 0029 236      .ENTRY MTH$GTAND, ^M<IV, R2, R3, R4, R5, R6, R7>
0028 237      : standard call-by-reference entry
0028 238      : disable DV (and FU), enable IV
0028 239      MTH$FLAG_JACKET
0028
6D 00000000'GF 9E 0028
0032      MOVAB G^MTH$$JACKET_HND, (FP)
0032      : set handler address to jacket
0032      : handler
0032
50 04 BC 50FD 0032 240      MOVG @X(AP), R0      ; R0/R1 = argument
01 10 0037 241      BSBB MTH$GTAND_R7      ; Get the tangent
04 0039 242      RET      ; With result in R0-R1
```

```
003A 244 .SBTTL MTH$GTAND_R7 - JSB entry point
003A 245 :
003A 246 :++
003A 247 :FUNCTIONAL DESCRIPTION:
003A 248 :
003A 249 :GTAND - JSB entry point
003A 250 :
003A 251 :Algorithmic steps:
003A 252 : 1. Make sure that the absolute value of the argument is greater than
003A 253 : 180/pi*2**-1024, otherwise GSIND will underflow.
003A 254 : 2. Compute GSIND and GCOSD.
003A 255 : 3. If GCOSD is zero, we have an error.
003A 256 : 4. Return GSIND / GCOSD.
003A 257 :
003A 258 :CALLING SEQUENCE:
003A 259 :
003A 260 :      MOVG      argument, R0
003A 261 :      JSB      MTH$GTAND_R7
003A 262 :
003A 263 :INPUT PARAMETERS:
003A 264 :
003A 265 :      R0 / R1 contains x
003A 266 :
003A 267 :IMPLICIT INPUTS:
003A 268 :
003A 269 :      NONE
003A 270 :
003A 271 :OUTPUT PARAMETERS:
003A 272 :
003A 273 :      The result is the tangent of x, in G_floating.
003A 274 :
003A 275 :IMPLICIT OUTPUTS:
003A 276 :
003A 277 :      NONE
003A 278 :
003A 279 :SIDE EFFECTS:
003A 280 :
003A 281 :      NONE
003A 282 :
003A 283 :--
003A 284 MTH$GTAND_R7::
003A 285      MOVG      R0, R2
003E 286      BICW      #^X8000, R2
0043 287      CMPW      #^X70, R2
0048 288      BLEQ      20$
004A 289      CMPG      G_SMALLEST_DEG, R2
004F 290      BLEQ      20$
0051 291      TSTG      R2
0054 292      BNEQ      UNFL
0056 293      CLRG      R0
0058 294      RSB
0059 295 20$:
0059 296      JSB      G^MTH$GSINCOSD_R7
005F 297      TSTG      R2
0062 298      BEQL      COSZER
0064 299      DIVG2     R2, R0
0068 300      RSB

52      52      50      50FD
52      8000      8F      AA
52      0070      8F      B1
          0F      15
52      B2      AF      51FD
          08      15
          52      53FD
          13      12
          50      7C
          05
00000000'GF      16
          52      53FD
          29      13
          50      52      46FD
          05
```

```
: Save argument
: R2/R3 = |argument|
: Compare |arg| with 2**-1017
: No possible underflow, compute GCOSD
: Possible underflow, use better test
: No possible underflow, compute GCOSD
: Check for arg = 0
: Branch to underflow logic if not zero
: Load R0/R1 with 0
: Return with value equal zero

: Compute GSIND(X) and GCOSD(X).
: Is GCOSD(X) zero ?
: If yes, error
: Compute GSIND / GCOSD
: Return.
```


MTH\$GTAN
1-011

; G Floating Point Tangent routine G 8
MTH\$GTAND_R7 - JSB entry point

0069 301

16-SEP-1984 01:32:14 VAX/VMS Macro V04-00
6-SEP-1984 11:24:22 [MTHRTL.SRC]MTHGTAN.MAR;1

Page 8
(7)

**F

```
0069 303 :  
0069 304 : COMMON ERROR CODE  
0069 305 :  
0069 306 :  
0069 307 :  
0069 308 : Underflow; if user has FU set, signal error. Always return 0.0  
0069 309 :  
0069 310 UNFL:  
00000000'GF 52 DC 0069 311 MOVPSL R2 ; R2 = user's or jacket routine's PSL  
04 00 FB 006B 312 CALLS #0, G^MTH$$JACKET_TST ; R0 = TRUE if JSB from jacket routine  
52 04 50 E9 0072 313 BLBC R0, 10$ ; branch if user did JSB  
AD 3C 0075 314 MOVZWL SF$W_SAVE_PSW(FP), R2 ; get user PSL saved by CALL  
50 D4 0079 315 10$: CLRL R0 ; R0 = result. LIB$SIGNAL will save in  
0D 52 06 E1 007B 316 ; CHF$L_MCH_R0/R1 so any handler can fixup  
6E DD 007F 317 BBC #6, R2, 20$ ; has user enabled floating underflow?  
7E 00'8F 9A 0081 318 PUSHL (SP) ; yes, return PC from special routine  
0085 319 MOVZBL #MTH$K_FLOUNDMAT, -(SP) ; trap code for hardware floating underflow  
0085 320 ; convert to MTH$_FLOUNDMAT (32-bit VAX-11  
0085 321 ; exception code)  
00000000'GF 02 FB 0085 322 CALLS #2, G^MTH$$SIGNAL ; signal (condition, PC)  
05 008C 323 20$: RSB ; return  
008D 324 :  
008D 325 :+  
008D 326 : COS was zero, so TAN is infinite, signal an error.  
008D 327 :-  
008D 328 COSZER:  
7E 00'8F 6E DD 008D 329 PUSHL (SP) ; Push "caller" PC  
50 01 0F 9A 008F 330 MOVZBL #MTH$K_FLOOVEMAT, -(SP) ; Condition value  
00000000'GF 02 FB 0093 331 ASHQ #15, #T, R0 ; R0/R1 = reserved operand  
05 0097 332 CALLS #2, G^MTH$$SIGNAL ; Signal an error  
009E 333 RSB ; Return to caller  
009F 334 :  
009F 335 .END
```


MTH\$GTAN
Symbol table

I 8
; G Floating Point Tangent routine

16-SEP-1984 01:32:14 VAX/VMS Macro V04-00
6-SEP-1984 11:24:22 [MTHRTL.SRC]MTHGTAN.MAR;1

Page 10
(8)

COSZER	0000008D	R	02
G_SMALLEST_DEG	00000000	R	02
MTH\$JACKET_HND	*****	X	02
MTH\$JACKET_TST	*****	X	00
MTH\$SIGNAL	*****	X	00
MTH\$GSINCOSD_R7	*****	X	00
MTH\$GSINCOS_R7	*****	X	00
MTH\$GTAN	00000008	RG	02
MTH\$GTAND	00000029	RG	02
MTH\$GTAND_R7	0000003A	RG	02
MTH\$GTAN_R7	00000019	RG	02
MTH\$K_FLOOVEMAT	*****	X	00
MTH\$K_FLOUNDMAT	*****	X	00
SFSW_SAVE_PSW =	00000004		
UNFL	00000069	R	02
X	= 00000004		

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
_MTH\$CODE	0000009F (159.)	02 (2.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	33	00:00:00.09	00:00:01.23
Command processing	120	00:00:00.71	00:00:04.90
Pass 1	120	00:00:01.43	00:00:06.36
Symbol table sort	0	00:00:00.02	00:00:00.03
Pass 2	72	00:00:00.76	00:00:02.79
Symbol table output	3	00:00:00.02	00:00:00.04
Psect synopsis output	2	00:00:00.02	00:00:00.05
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	352	00:00:03.06	00:00:15.40

The working set limit was 1050 pages.
6467 bytes (13 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 44 non-local and 3 local symbols.
395 source lines were read in Pass 1, producing 16 object records in Pass 2.
9 pages of virtual memory were used to define 8 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	4

MTHSGTAN
VAX-11 Macro Run Statistics

; G Floating Point Tangent routine J 8

16-SEP-1984 01:32:14 VAX/VMS Macro V04-00
6-SEP-1984 11:24:22 [MTHRTL.SRC]MTHGTAN.MAR;1

Page 11
(8)

88 GETS were required to define 4 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LISS:MTHGTAN/OBJ=OBJ\$:MTHGTAN MSRC\$:MTHJACKET/UPDATE=(ENHS:MTHJACKET)+MSRC\$:

MTH
1-0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY